Managed Host User Setup

For hosts that will be managed by Ansible, or some other automated method, the host will need a user account with proper access.

Follow these steps to setup the account and access.

> NOTE: This page assumes that SSH server is setup.
> And, that the firewall rules are setup for remote SSH access.

Provisioning User Account

Create User

Create a provisioning user on the host.

> NOTE: If not running as root, you will need to prepend these commands with 'sudo'.

> NOTE: By default, we use the username 'provisioner'.

Run this to create the user with a password:

```
useradd -m provisioner
```

> NOTE: We included the '-m' switch, to give the user a home directory.
> This is done, so that our user can authenticate with SSH keys, which need a .ssh folder under the user's home directory.

Once created, add the provisioner account to the sudo group, so it can run elevated commands:

```
usermod -aG sudo provisioner
```

> NOTE: The `-aG` option tells `usermod` to add the user to the listed groups.

User Shell

If you want the provisioner user to have shell access, set the shell with this:

```
sudo chsh -s /bin/bash provisioner
```

See this page for details: Linux: Shell Appearance

> NOTE: It may not be necessary for an automation account to have a shell.
> But, it does help when troubleshooting.

Account Password

You need to choose if you want the provisioner account to have a password or not.

> NOTE: You don't really have to create a password for a provisioning user account if:
> - You will use SSH keys for authentication,
> - You will be bypassing sudo checks for this user

If you want it to have a password, use this to create one:

```
passwd provisioner
```

If you don't want the user to have a password, run this:

```
passwd -l provisioner
```

The above will lock the password, making login via password impossible.

Sudo Bypass

We will allow the provisioner user to skip any sudo challenge checks.
We do this, so that we don't have to manage passwords in the deployment scripting.

Follow this page for how to do that: Linux: Allow User to Skip Sudo Challenge

Edit the visudo file with this:

```
visudo -f /etc/sudoers
```

And, add this line above the includedir line:

```
provisioner ALL=(ALL) NOPASSWD: ALL
```

SSH Key Installation

Install an SSH key for the user.

> NOTE: If you are just setting up this user, it may not yet have an authorized_keys file, or an .ssh folder in its profile.

If the .ssh folder is missing, follow this: Linux: Missing .SSH Folder

To do so, locate the authorized_keys file for the provisioner user at:
/home/provisioner/.ssh/authorized_keys.
And, add the public key of the provisioner account.

Open the authorized_keys file for the provisioner's account, with this:

```
sudo nano /home/provisioner/.ssh/authorized_keys
```

Paste in the public key string of the provisioner account's SSH public key.

The current provisioner account SSH public key is found here:

- On Build Server:
  "/mnt/secshare/oga/keys/provisioner_user/ecdsa-key-provisioner-buildserver01-20250803.pub"
- In Secure Share:
  "\SecureShare git\oga\keys\provisioner_user"

Save and close when done.

Verify SSH Access

Once the SSH key is loaded, you can verify SSH authentication with this:

```
ssh -i /path/to/privatekeyfile provisioner@hostname
```

To test access with the provisioner's current SSH key file, use something like this:

```
ssh -i /mnt/secshare/oga/keys/provisioner_user/ecdsa-key-provisioner-buildserver01-20250803.key provisioner@hostname
```

Update Ansible Hosts File

For Ansible to reach the host, you need to give Ansible some information about it.

Live Ansible Hosts File

This is stored in the Ansible hosts file.

The live hosts file is on the Ansible server at: /etc/ansible/hosts

Add the host to it, like this:

```
[Name of host group]
hostname ansible_host=192.168.120.99 ansible_user=provisioner
ansible_ssh_private_key_file=/mnt/secshare/oga/keys/provisioner_user/ecdsa-key-provisioner-buildserver01-20250803.key
```

The above is a host entry in the Ansible hosts file.

It has the following parts, in this order:

- Hostname - This is name used by ansible plays to identify the host.
- ansible_user - This is the username that Ansible will login with. Set it to the account name, from above.
- ansible_ssh_private_key_file - This is the path to the current SSH private key used for deployment.

The current provisioning private key file, for Ansible, is in the mapped secure share, here:

```
/mnt/secshare/oga/keys/provisioner_user/ecdsa-key-provisioner-buildserver01-20250803.key
```

Org Hosts File

As well, there's a hosts file in each Org's Infrastructure project, as well.

This reflects what ansible host entries are relevant to that org.

Copy the same host entry into that file, to keep it in sync.

Revision #1
Created 3 August 2025 20:18:23 by glwhite
Updated 3 August 2025 20:34:36 by glwhite